Six-dimensional quantum dynamics of H2 dissociative adsorption on the Pt(211) stepped surface

Olsen, R.A.; McCormack, D.A.; Luppi, M.; Baerends, E.J.

published in
Journal of Chemical Physics
2008

DOI (link to publisher)
10.1063/1.2920488

document version
Publisher's PDF, also known as Version of record

An efficient algorithm for solving nonlinear equations with a minimal number of trial vectors: Applications to atomic-orbital based coupled-cluster theory

Marcin Ziółkowski,[1,a)] Ville Weijo,[1,2] Poul Jørgensen,[1] and Jeppe Olsen[1]
[1]*The Lundbeck Foundation Center for Theoretical Chemistry, Department of Chemistry, University of Aarhus, Langelandsgade 140, DK-8000 Århus C, Denmark*
[2]*Laboratory of Physics, Helsinki University of Technology, P.O. Box 1100 (Otakaari 1 M), 02015 Espoo, Finland*

(Received 22 February 2008; accepted 23 April 2008; published online 23 May 2008)

The conjugate residual with optimal trial vectors (CROP) algorithm is developed. In this algorithm, the optimal trial vectors of the iterations are used as basis vectors in the iterative subspace. For linear equations and nonlinear equations with a small-to-medium nonlinearity, the iterative subspace may be truncated to a three-dimensional subspace with no or little loss of convergence rate, and the norm of the residual decreases in each iteration. The efficiency of the algorithm is demonstrated by solving the equations of coupled-cluster theory with single and double excitations in the atomic orbital basis. By performing calculations on H_2O with various bond lengths, the algorithm is tested for varying degrees of nonlinearity. In general, the CROP algorithm with a three-dimensional subspace exhibits fast and stable convergence and outperforms the standard direct inversion in iterative subspace method. © *2008 American Institute of Physics*. [DOI: 10.1063/1.2928803]

I. INTRODUCTION

The solution of linear and nonlinear equations is a central task of electronic structure theory. For example, in algorithms for diagonalization free optimization of the Hartree–Fock and Kohn–Sham energies linear[1] or nonlinear[2] equations are solved, whereas in coupled-cluster theory[3] nonlinear equations are solved.

The solution of linear equations with a symmetric and positive definite matrix may be formulated as the minimization of a quadratic function. The standard method for this minimization is the conjugate gradient (CG) method.[4–7] In each iteration of the CG algorithm, a new direction is added to the previous directions. By ensuring that the new direction is conjugate to the previous directions, the CG algorithm obtains the attractive feature that a simple unidirectional minimization is mathematically equivalent to a minimization in the space spanned by all directions generated in the current and previous iterations. Further, only information from the last iteration is needed to identify the unidirectional search direction. The storage and manipulation of all generated directions are therefore avoided without loss of convergence. In quantum chemistry, Pople *et al.*[8] used an iterative subspace method for solving the coupled perturbed Hartree–Fock equations which Wormer *et al.*[9] showed was a special implementation of the CG method. If the matrix defining the linear equations is nonsymmetric, the CG method cannot be applied in its standard form, and if the matrix is symmetric but not positive definite, the CG algorithm in its standard form is not guaranteed to be able to determine the solution. The CG method has been reformulated to forms that are more suitable for symmetric matrices that are not positive definite.[7,10] However, for matrices that are nonsymmetric or not positive definite, it may be more attractive to use minimal residual (MR) methods.[7] For general nonsymmetric matrices, the generalized minimal residual method is a standard choice.[11] For symmetric matrices, the conjugate residual (CR) method[12] shares the very attractive property with the CG method that each iteration can be expressed in terms of a unidirectional search, where the search direction may be determined from information from the last iteration, and where the storage and manipulation of a long list of directions and residuals therefore is avoided. A number of variants of the CR methods have been developed for the nonsymmetric matrices.[11,13,14] We refer to numerical mathematical texts for further detail.[7]

In quantum chemistry, linear and nonlinear equations are often solved by using the direct inversion of iterative subspace (DIIS) method.[15,16] The DIIS method was originally developed to improve the local convergence of self-consistant field calculations, but it has proven useful for the solution of many other problems in electronic structure theory including geometry optimization[17,18] and the solution of the coupled-cluster equations.[19] A short review on the properties and use of the DIIS method has recently been published.[20] In each iteration of the DIIS algorithm, a residual is minimized in the subspace of the current and previous trial vectors. As the dimension of the trial vectors may be large, it is often only possible to store the information from the last few iterations. However, in contrast to the CG and CR methods, a reduction in the rate of convergence is typically observed when vectors from earlier iterations are discarded, even when linear equations with symmetric matri-

[a)]Author to whom correspondence should be addressed. Electronic mail: marcin@chem.au.dk.

ces are solved. There are thus important differences between the DIIS method and the standard MR algorithms which will be analyzed in this paper.

The CR algorithm will be derived and reparametrized to a form where trial solutions rather than directions are stored. The resulting algorithm, the CR algorithm with Optimal trial vectors (CROP), is mathematically equivalent to the CR algorithm for linear equations but is straightforward to extend to nonlinear equations. For nonlinear equations with a small nonlinearity only a minor degradation in the performance is seen using the CROP algorithm storing only the last trial solution, as the degradation is due solely to the nonlinearity of the equations. For linear equations, we show that the CROP method with the last trial solution stored gives the same solution as the standard DIIS algorithm where information from all previous iterations are stored.

The CROP method is a general method for solving linear and nonlinear equations. The efficiency of the method is demonstrated and compared to the one of the standard DIIS method for linear equations solving orthogonalized atomic orbital second order Møller–Plesset[21] (MP2) equations and for nonlinear equations solving orthogonalized atomic orbital coupled-cluster singles doubles[22] (CCSD) equations.

II. MINIMAL RESIDUAL METHODS FOR LINEAR EQUATIONS

In this section, we discuss and compare various minimal residual algorithms for determining the solution $\mathbf{x}^\star$ of the linear equation

$$\mathbf{A}\mathbf{x}^\star - \mathbf{b} = 0, \tag{1}$$

where $\mathbf{A}$ has a dimension d and is symmetric but not necessarily positive definite. We will assume that $\mathbf{A}$ is nonsingular, although the following derivation also holds if $\mathbf{A}$ is singular and $\mathbf{b}$ is orthogonal to the space of the null space of $\mathbf{A}$, i.e., the space of eigenvectors of $\mathbf{A}$ with eigenvalues equal to zero. The residual for a general vector $\mathbf{x}$ is given as

$$\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x} \tag{2}$$

and may be used as a measure of the accuracy of the solution. Minimizing the squared residual norm

$$g(\mathbf{x}) = \mathbf{r}^T\mathbf{r} \tag{3}$$

gives

$$\frac{\partial g(\mathbf{x})}{\partial \mathbf{x}} = 2\mathbf{A}(\mathbf{A}\mathbf{x} - \mathbf{b}) = 0. \tag{4}$$

Minimization of $g(\mathbf{x})$ may therefore be used to determine the solution of Eq. (1).

A. The conjugate residual algorithm

1. Introduction

In this section, we derive the CR algorithm. We first present the CR algorithm in its standard form[7] and then reparametrize it to a form where the optimal solution of each iteration rather than the optimal direction is stored.

In iteration $n+1$ of the CR algorithm, with the approximate solution $\mathbf{x}_n$, the residual $\mathbf{r}_n = \mathbf{b} - \mathbf{A}\mathbf{x}_n$ is first calculated and the trial solution $\mathbf{x}_{n+1}$ is parametrized as a linear combination of the residual $\mathbf{r}_n$ and the optimal search directions of the previous iterations $\{\mathbf{p}\}, i = 0, 1, \ldots, n-1$

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \sum_{i=0}^{n-1} \alpha_i^{(n)}\mathbf{p}_i + \alpha_n^{(n)}\mathbf{r}_n. \tag{5}$$

The idea of the CR algorithm is to identify an optimal search direction $\mathbf{p}_n$ replacing the multiple search directions in Eq. (5)

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \alpha_n^{(n)}\mathbf{p}_n \tag{6}$$

such that the minimization of the norm of the residual $g(\mathbf{x}_{n+1})$ with $\mathbf{x}_{n+1}$ calculated either by Eq. (5) or Eq. (6) gives a mathematically identical result. In the CR algorithm, as in the more commonly used CG method, all but the last direction may be discarded without loss of convergence rate. We will now describe the CR algorithm in a fashion that is in accordance with the standard quantum chemical focus on subspace optimizations. We will, in particular, show how the optimal search directions may be obtained such that the multidirectional search of Eq. (5) is identical to the unidirectional search of Eq. (6).

2. Iteration 1

Let $\mathbf{x}_0$ be our starting guess. The initial iteration differs from the remaining iterations by having a single search direction $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$, which, therefore, is optimal, i.e., $\mathbf{p}_0 = \mathbf{r}_0$. The solution vector at iteration 1, therefore, has the form

$$\mathbf{x}_1 = \mathbf{x}_0 + \alpha_0^{(0)}\mathbf{p}_0. \tag{7}$$

Minimization of $g(\mathbf{x}_1)$ in the direction $\mathbf{p}_0$ gives

$$\frac{\partial g(\mathbf{x}_1)}{\partial \alpha_0^{(0)}} = \alpha_0^{(0)} 2\mathbf{p}_0^T\mathbf{A}^2\mathbf{p}_0 - 2\mathbf{r}_0^T\mathbf{A}\mathbf{p}_0 = 0, \tag{8}$$

which determines the optimal steplength in the direction $\mathbf{p}_0$,

$$\alpha_0^{(0)} = \frac{\mathbf{r}_0^T\mathbf{A}\mathbf{p}_0}{\mathbf{p}_0^T\mathbf{A}^2\mathbf{p}_0}. \tag{9}$$

Note that $\mathbf{p}_0^T\mathbf{A}^2\mathrm{p}_0$ is nonvanishing as we have assumed that $\mathbf{A}$ is nonsingular. The residual at $\mathbf{x}_1$,

$$\mathbf{r}_1 = \mathbf{b} - \mathbf{A}\mathbf{x}_1, \tag{10}$$

is shown using Eqs. (9) and (7) to be conjugate to the search direction $\mathbf{p}_0$,

$$\mathbf{p}_0^T\mathbf{A}\mathbf{r}_1 = \mathbf{r}_0^T\mathbf{A}\mathbf{r}_1 = 0. \tag{11}$$

Further, from Eq. (7), we see that

$$\mathbf{A}\mathbf{p}_0 = \frac{1}{\alpha_0^{(0)}}(\mathbf{r}_0 - \mathbf{r}_1). \tag{12}$$

3. Iteration 2

In iteration 2, the trial vector is initially written as having components along $\mathbf{p}_0$ and $\mathbf{r}_1$,

$$\mathbf{x}_2 = \mathbf{x}_1 + \alpha_0^{(1)}\mathbf{p}_0 + \alpha_1^{(1)}\mathbf{r}_1. \tag{13}$$

Minimization of $g(\mathbf{x}_2)$ with respect to $\alpha_0^{(1)}$ and $\alpha_1^{(1)}$ gives the subspace equations

$$\begin{pmatrix} \mathbf{p}_0^T\mathbf{A}^2\mathbf{p}_0 & \mathbf{p}_0^T\mathbf{A}^2\mathbf{r}_1 \\ \mathbf{p}_0^T\mathbf{A}^2\mathbf{r}_1 & \mathbf{r}_1^T\mathbf{A}^2\mathbf{r}_1 \end{pmatrix}\begin{pmatrix} \alpha_0^{(1)} \\ \alpha_1^{(1)} \end{pmatrix} = \begin{pmatrix} 0 \\ \mathbf{r}_1^T\mathbf{A}\mathbf{r}_1 \end{pmatrix}. \tag{14}$$

From the first row of Eq. (14), the coefficient $\alpha_0^{(1)}$ may be written in terms of $\alpha_1^{(1)}$,

$$\alpha_0^{(1)} = -\alpha_1^{(1)}\frac{\mathbf{p}_0^T\mathbf{A}^2\mathbf{r}_1}{\mathbf{p}_0^T\mathbf{A}^2\mathbf{p}_0}. \tag{15}$$

The trial vector of Eq. (13) minimizing the residual norm may therefore be expressed as a unidirectional search

$$\mathbf{x}_2 = \mathbf{x}_1 + \alpha_1^{(1)}\mathbf{p}_1, \tag{16}$$

where the direction $\mathbf{p}_1$ is

$$\mathbf{p}_1 = \mathbf{r}_1 - \frac{\mathbf{p}_0^T\mathbf{A}^2\mathbf{r}_1}{\mathbf{p}_0^T\mathbf{A}^2\mathbf{p}_0}\mathbf{p}_0. \tag{17}$$

Having $\mathbf{x}_2$ simplified to a unidirectional search, the minimization of $g(\mathbf{x}_2)$ may proceed as in iteration 1 [see Eq. (7)]. In analogy to Eq. (9), the optimal steplength in the direction $\mathbf{p}_1$ therefore becomes

$$\alpha_1^{(1)} = \frac{\mathbf{r}_1^T\mathbf{A}\mathbf{p}_1}{\mathbf{p}_1^T\mathbf{A}^2\mathbf{p}_1}, \tag{18}$$

and $\mathbf{x}_2$ has thus been determined. The residual at $\mathbf{x}_2$ becomes

$$\mathbf{r}_2 = \mathbf{b} - \mathbf{A}\mathbf{x}_2, \tag{19}$$

or using Eq. (16)

$$\mathbf{r}_2 = \mathbf{r}_1 - \alpha_1^{(1)}\mathbf{A}\mathbf{p}_1. \tag{20}$$

There are a number of relations that will be used when deriving the subspace equations for the following iterations. Multiplying Eq. (17) with $\mathbf{p}_0^T\mathbf{A}^2$ gives

$$\mathbf{p}_0^T\mathbf{A}^2\mathbf{p}_1 = 0. \tag{21}$$

From Eqs. (20) and (18), one obtains

$$\mathbf{r}_2^T\mathbf{A}\mathbf{p}_1 = 0, \tag{22}$$

and Eqs. (20), (11), and (21) give

$$\mathbf{r}_2^T\mathbf{A}\mathbf{p}_0 = \mathbf{r}_2^T\mathbf{A}\mathbf{r}_0 = 0. \tag{23}$$

By using Eq. (17) to obtain an expression for $\mathbf{r}_1$ and using Eqs. (22) and (23), one obtains

$$\mathbf{r}_2^T\mathbf{A}\mathbf{r}_1 = 0. \tag{24}$$

Finally, by expressing $\mathbf{A}\mathbf{p}_0$ in the form of Eq. (12) and using Eqs. (23) and (24), one obtains

$$\mathbf{r}_2^T\mathbf{A}^2\mathbf{p}_0 = 0. \tag{25}$$

4. Iteration $n+1$

Let us now consider iteration $n+1$. The previous directions and residuals fulfill the relations

$$\mathbf{r}_i^T\mathbf{A}\mathbf{p}_j = 0, \quad i,j = 0,1,\ldots,n, \quad i > j, \tag{26}$$

$$\mathbf{r}_i^T\mathbf{A}\mathbf{r}_j = 0, \quad i,j = 0,1,\ldots,n, \quad i > j, \tag{27}$$

$$\mathbf{r}_i^T\mathbf{A}^2\mathbf{p}_j = 0, \quad i,j = 0,1,\ldots,n, \quad i > j+1, \tag{28}$$

$$\mathbf{p}_i^T\mathbf{A}^2\mathbf{p}_j = \mathbf{p}_i^T\mathbf{A}^2\mathbf{p}_j\delta_{ij}, \quad i,j = 0,1,\ldots,n-1. \tag{29}$$

The new trial vector is initially written as a general vector in the space spanned by the previous search directions $\{\mathbf{p}_i\}, i = 0,1,\ldots,n-1$ and the current residual $\mathbf{r}_n$

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \sum_{i=0}^{n-1}\alpha_i^{(n)}\mathbf{p}_i + \alpha_n^{(n)}\mathbf{r}_n. \tag{30}$$

Minimizing $g(\mathbf{x}_{n+1})$ with respect to the $n+1$ free parameters and using Eqs. (26), (28), and (29), one obtains the subspace equations

$$\begin{pmatrix} \widetilde{\mathbf{A}^2} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{p}_{n-1}^T\mathbf{A}^2\mathbf{p}_{n-1} & \mathbf{p}_{n-1}^T\mathbf{A}^2\mathbf{r}_n \\ \mathbf{0} & \mathbf{p}_{n-1}^T\mathbf{A}^2\mathbf{r}_n & \mathbf{r}_n^T\mathbf{A}^2\mathbf{r}_n \end{pmatrix}\alpha^{(n)} = \begin{pmatrix} \mathbf{0} \\ 0 \\ \mathbf{r}_n^T\mathbf{A}\mathbf{r}_n \end{pmatrix}, \tag{31}$$

where elements of matrix $\widetilde{\mathbf{A}^2}$ are defined as

$$\widetilde{\mathbf{A}^2}_{ij} = \mathbf{p}_i^T\mathbf{A}^2\mathbf{p}_j\delta_{ij}, \quad i,j = 0,1,\ldots,n-2. \tag{32}$$

Due to the form of the subspace equations, $\mathbf{x}_{n+1}$ may be expressed in terms of a single search direction $\mathbf{p}_n$ and of the optimal steplength $\alpha_n^{(n)}$ as

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \alpha_n^{(n)}\mathbf{p}_n, \tag{33}$$

where

$$\mathbf{p}_n = \mathbf{r}_n - \frac{\mathbf{p}_{n-1}^T\mathbf{A}^2\mathbf{r}_n}{\mathbf{p}_{n-1}^T\mathbf{A}^2\mathbf{p}_{n-1}}\mathbf{p}_{n-1}, \tag{34}$$

$$\alpha_n^{(n)} = \frac{\mathbf{r}_n^T\mathbf{A}\mathbf{p}_n}{\mathbf{p}_n^T\mathbf{A}^2\mathbf{p}_n}. \tag{35}$$

The residual of $\mathbf{x}_{n+1}$ is then obtained as

$$\mathbf{r}_{n+1} = \mathbf{r}_n - \alpha_n^{(n)}\mathbf{A}\mathbf{p}_n. \tag{36}$$

Equations (26)–(29) are valid when n is increased by 1 and the iterative procedure of the CR algorithm is therefore established by induction.[23]

In iteration n, we need to store the following vectors: $\mathbf{x}_n$, $\mathbf{p}_{n-1}$, $\mathbf{A}\mathbf{p}_{n-1}$, and $\mathbf{r}_n$. CR method requires the storage of one vector, $\mathbf{A}\mathbf{p}$, more than the CG method. In iteration $n+1$, the linear transformation $\mathbf{A}\mathbf{r}_n$ may be carried out, and $\mathbf{x}_{n+1}$, $\mathbf{p}_n$, $\mathbf{A}\mathbf{p}_n$ may be obtained by using Eqs. (33)–(35). Finally, $\mathbf{r}_{n+1}$ may be obtained by using Eq. (36). Each iteration represents a minimization of the residual in a subspace containing the subspace of the previous iteration with one new direction added and the residual therefore decreases in each iteration.

It is seen from Eq. (29) that the directions $\mathbf{p}_i$ constitute a set of linear independent vectors. In iteration d, one is therefore minimizing the residual in the full d-dimensional vector space, and the exact solution $\mathbf{x}^\star$ is therefore obtained in this iteration. In exact arithmetic, the CR algorithm therefore converges to the exact solution in a number of iterations

given by the dimension of the matrix. If iteration $d+1$ is started, the residual $\mathbf{r}_{d+1}$ is vanishing, so Eqs. (35) and (34) cannot be used to obtain a direction $\mathbf{p}_{d+1}$ that is conjugate to $\mathbf{p}_d$. When the CR method is used, usually in a preconditioned form (see Sec. II C), a good approximation to the exact solution is obtained in a number of iterations that is much smaller than the dimension of the matrix. In Sec. IV, we will discuss the rate of convergence for the CG and CR methods.

B. The conjugate residual with optimal trial vectors algorithm

In the CR algorithm described in the previous subsection, iteration $n+1$ may be viewed as a minimization of the residual norm in the space spanned by the $\mathbf{p}_i, i=0,1,\ldots,n-1$ and $\mathbf{r}_n$. As the norm of a vector is independent of the basis in which it is expressed, any other basis that spans the same space produces the same trial solution $\mathbf{x}_{n+1}$. There is therefore considerable flexibility in the way search directions may be chosen. We consider a choice—the CROP method—where a generalization of the CR method to nonlinear equations is straightforward.

Consider a space of the form of Eq. (30) but restrict the space to contain only the residual $\mathbf{r}_n$ and the last $n-k$ search directions with $0 \leqslant k \leqslant n-1$,

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \sum_{i=k}^{n-1} \alpha_i^{(n)} \mathbf{p}_i + \alpha_n^{(n)} \mathbf{r}_n. \tag{37}$$

From the residual $\mathbf{r}_n$, we further introduce a preliminary improvement of $\mathbf{x}_n$ as

$$\tilde{\mathbf{x}}_{n+1} = \mathbf{x}_n + \mathbf{r}_n. \tag{38}$$

Using Eq. (33) for a general index

$$\mathbf{p}_i = \frac{\mathbf{x}_{i+1} - \mathbf{x}_i}{\alpha_i^{(i)}} \tag{39}$$

gives

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \sum_{i=k}^{n-1} \frac{\alpha_i^{(n)}}{\alpha_i^{(i)}} (\mathbf{x}_{i+1} - \mathbf{x}_i) + \alpha_n^{(n)} (\tilde{\mathbf{x}}_{n+1} - \mathbf{x}_n), \tag{40}$$

which may be written as

$$\begin{aligned} \mathbf{x}_{n+1} &= \mathbf{x}_n + \sum_{i=k}^{n-1} \frac{\alpha_i^{(n)}}{\alpha_i^{(i)}} (\mathbf{x}_{i+1} - \mathbf{x}_n + \mathbf{x}_n - \mathbf{x}_i) + \alpha_n^{(n)} (\tilde{\mathbf{x}}_{n+1} - \mathbf{x}_n) \\ &= \mathbf{x}_n + \sum_{i=k}^{n-1} c_i (\mathbf{x}_i - \mathbf{x}_n) + c_{n+1} (\tilde{\mathbf{x}}_{n+1} - \mathbf{x}_n), \end{aligned} \tag{41}$$

where

$$c_i = \begin{cases} \alpha_n^{(n)} & \text{if } i = n+1 \\ \dfrac{\alpha_{i-1}^{(n)}}{\alpha_{i-1}^{(i-1)}} - \dfrac{\alpha_i^{(n)}}{\alpha_i^{(n)}} & \text{if } k < i < n \\ -\dfrac{\alpha_k^{(n)}}{\alpha_k^{(k)}} & \text{if } i = k. \end{cases} \tag{42}$$

Note that no coefficient c_n is defined. The space spanned by $\mathbf{r}_n$ and the last $n-k$ search directions is thus identical to the space spanned by the $\tilde{\mathbf{x}}_{n+1} - \mathbf{x}_n$ and $\mathbf{x}_i - \mathbf{x}_n, i=k,k+1,\ldots,n-1$.

For $k=0$, we have that the minimization of the residual norm is carried out in the space

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \sum_{i=0}^{n-1} c_i (\mathbf{x}_i - \mathbf{x}_n) + c_{n+1} (\tilde{\mathbf{x}}_{n+1} - \mathbf{x}_n), \tag{43}$$

giving an approximate solution $\mathbf{x}_{n+1}$ that is identical to the one obtained in the CR method. Furthermore, as we have shown that the optimal trial vector of iteration $n+1$ is contained in the space spanned by $\mathbf{r}_n$ and $\mathbf{p}_{n-1}$, the optimal trial vector is contained in the space spanned by $\tilde{\mathbf{x}}_{n+1} - \mathbf{x}_n$ and $\mathbf{x}_{n-1} - \mathbf{x}_n$,

$$\mathbf{x}_{n+1} = \mathbf{x}_n + c_{n-1} (\mathbf{x}_{n-1} - \mathbf{x}_n) + c_{n+1} (\tilde{\mathbf{x}}_{n+1} - \mathbf{x}_n). \tag{44}$$

We denote the parametrization of the CR algorithm, where OPtimal trial vectors are used as the CROP algorithm, and discuss now the equations that may be used to determine the coefficients c_i of Eq. (43). The residual corresponding to Eq. (43) may be written as

$$\mathbf{r}_{n+1} = \mathbf{r}_n + \sum_{i=0}^{n} c_i (\mathbf{r}_i - \mathbf{r}_n) + c_{n+1} (\tilde{\mathbf{r}}_{n+1} - \mathbf{r}_n), \tag{45}$$

where

$$\tilde{\mathbf{r}}_{n+1} = \mathbf{b} - \mathbf{A}\tilde{\mathbf{x}}_{n+1}. \tag{46}$$

When the c-parametrization in Eq. (43) is used, the minimization of the norm of $\mathbf{r}_{n+1}$ in Eq. (45) requires that a small set of linear equations are solved. Minimizing $\|\mathbf{r}_{n+1}\|^2$ with respect to c_i gives the set of linear equations of dimension $n+1$

$$\mathbf{D}\mathbf{c} = \mathbf{h}, \tag{47}$$

where

$$D_{ij} = \begin{cases} (\mathbf{r}_i - \mathbf{r}_n)^T (\mathbf{r}_j - \mathbf{r}_n) & \text{if } i,j \leqslant n \\ (\mathbf{r}_i - \mathbf{r}_n)^T (\tilde{\mathbf{r}}_n - \mathbf{r}_n) & \text{if } i \leqslant n, j = n+1 \\ (\tilde{\mathbf{r}}_n - \mathbf{r}_n)^T (\mathbf{r}_j - \mathbf{r}_n) & \text{if } i = n+1, j \leqslant n \\ (\tilde{\mathbf{r}}_n - \mathbf{r}_n)^T (\tilde{\mathbf{r}}_n - \mathbf{r}_n) & \text{if } i = j = n+1, \end{cases} \tag{48}$$

$$h_i = \begin{cases} -(\mathbf{r}_i - \mathbf{r}_n)^T \mathbf{r}_n & \text{if } i \leqslant n \\ -(\tilde{\mathbf{r}}_n - \mathbf{r}_n)^T \mathbf{r}_n & \text{if } i = n+1. \end{cases} \tag{49}$$

The equations for determining the coefficients $\mathbf{c}$ may be solved in an alternative fashion which emphasizes the similarity to the DIIS method of Pulay.[15,16] By introducing in Eq. (43), the coefficient

$$c_n = 1 - \sum_{i=0}^{n-1} c_i - c_{n+1}, \tag{50}$$

a symmetric expression is obtained

$$\mathbf{x}_{n+1} = \sum_{i=0}^{n} c_i \mathbf{x}_i + c_{n+1} \tilde{\mathbf{x}}_{n+1}, \tag{51}$$

where the parameters c_i are constrained by the relation obtained from Eq. (50)

$$\sum_{i=0}^{n+1} c_i = 1, \tag{52}$$

and the expansion coefficients are determined by minimizing the residual norm

$$\mathbf{r}_{n+1} = \sum_{i=0}^{n} c_i \mathbf{r}_i + c_{n+1}\tilde{\mathbf{r}}_{n+1}. \tag{53}$$

The reduced expression for $\mathbf{x}_{n+1}$ of Eq. (44) may similarly be written as

$$\mathbf{x}_{n+1} = c_{n-1}\mathbf{x}_{n-1} + c_n\mathbf{x}_n + c_{n+1}\tilde{\mathbf{x}}_{n+1}, \tag{54}$$

with the constraint

$$c_{n-1} + c_n + c_{n+1} = 1, \tag{55}$$

where the expansion coefficients are determined by minimizing the norm of the residual

$$\mathbf{r}_{n+1} = c_{n-1}\mathbf{r}_{n-1} + c_n\mathbf{r}_n + c_{n+1}\tilde{\mathbf{r}}_{n+1}. \tag{56}$$

The minimization of the norm of residual of $\mathbf{x}_{n+1}$ in the parametrization of Eq. (51) with the constraint of Eq. (52) may be carried out as a standard DIIS determination of the expansion coefficients. Using an undetermined Lagrange multiplier λ to ensure the fulfillment of the constraint Eq. (52), the minimization is expressed in terms of the solution of a set of $n+1$ linear equations

$$\begin{pmatrix} \mathbf{B}^{\mathrm{CR}} & -\mathbf{1} \\ -\mathbf{1} & 0 \end{pmatrix}\begin{pmatrix} \mathbf{c} \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ -1 \end{pmatrix}, \tag{57}$$

where

$$B_{ij}^{\mathrm{CR}} = \begin{cases} \mathbf{r}_i^T\mathbf{r}_j & \text{if } i,j \leqslant n \\ \mathbf{r}_i^T\tilde{\mathbf{r}}_{n+1} & \text{if } i \leqslant n, j = n+1 \\ \tilde{\mathbf{r}}_{n+1}^T\mathbf{r}_j & \text{if } i = n+1, j \leqslant n \\ \tilde{\mathbf{r}}_{n+1}^T\tilde{\mathbf{r}}_{n+1} & \text{if } i = j = n+1. \end{cases} \tag{58}$$

C. The preconditioned conjugate residual algorithm

To improve convergence, it is a standard practice to introduce preconditioning, i.e., to introduce a coordinate transformation that produces a new set of equations with a matrix that has a lower condition number. To accomplish this transformation, the linear equations, Eq. (1), are multiplied with the transpose of a nonsingular matrix $\mathcal{P}$

$$\mathcal{P}^T\mathbf{A}\mathbf{x} - \mathcal{P}^T\mathbf{b} = 0. \tag{59}$$

Introducing the new set of coordinates $\mathbf{Y} = \mathcal{P}^{-1}\mathbf{x}$, Eq. (59) gives

$$\mathbf{A}^{\mathcal{P}}\mathbf{Y} - \mathbf{b}^{\mathcal{P}} = 0, \tag{60}$$

where

$$\mathbf{A}^{\mathcal{P}} = \mathcal{P}^T\mathbf{A}\mathcal{P}, \quad \mathbf{b}^{\mathcal{P}} = \mathcal{P}^T\mathbf{b}. \tag{61}$$

We may thus solve Eq. (60) by using the CR algorithm and then backtransform this solution to the original coordinates. An alternative approach is to solve Eq. (60) in the original basis using the modified CR equations. This may be done using Eq. (61), writing the residual in the $\mathbf{Y}$ basis as

$$\mathbf{r}^{\mathcal{P}} = \mathbf{b}^{\mathcal{P}} - \mathbf{A}^{\mathcal{P}}\mathbf{Y} = \mathcal{P}^T\mathbf{r}, \tag{62}$$

and introducing the matrix $\mathbf{C}$

$$\mathbf{C}^{-1} = \mathcal{P}\mathcal{P}^T. \tag{63}$$

The modified CR equations then read as

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \alpha_n^{(n)}\mathbf{C}^{-1}\mathbf{p}_n, \tag{64}$$

where

$$\mathbf{p}_n = \mathbf{r}_n - \frac{\mathbf{p}_{n-1}^T\mathbf{C}^{-1}\mathbf{A}\mathbf{C}^{-1}\mathbf{A}\mathbf{C}^{-1}\mathbf{r}_n}{\mathbf{p}_{n-1}^T\mathbf{C}^{-1}\mathbf{A}\mathbf{C}^{-1}\mathbf{A}\mathbf{C}^{-1}\mathbf{p}_{n-1}}\mathbf{p}_{n-1}, \tag{65}$$

$$\alpha_n^{(n)} = \frac{\mathbf{r}_n^T\mathbf{C}^{-1}\mathbf{A}\mathbf{C}^{-1}\mathbf{p}_n}{\mathbf{p}_n^T\mathbf{C}^{-1}\mathbf{A}\mathbf{C}^{-1}\mathbf{A}\mathbf{C}^{-1}\mathbf{p}_n}, \tag{66}$$

and

$$\mathbf{r}_{n+1} = \mathbf{r}_n - \alpha_n^{(n)}\mathbf{A}\mathbf{C}^{-1}\mathbf{p}_n. \tag{67}$$

Choosing $\mathcal{P}^T$ such that $\mathbf{C}^{-1}$ is a good approximation to $\mathbf{A}$ ensures that the linear equations are solved on a basis where the matrix $\mathbf{A}$ has a lower condition number.

In the preconditioned CR algorithm, we store in iteration n: $\mathbf{x}_n$, $\mathbf{C}^{-1}\mathbf{p}_{n-1}$, $\mathbf{A}\mathbf{C}^{-1}\mathbf{p}_{n-1}$, $\mathbf{C}^{-1}\mathbf{r}_n$, and carry out the $\mathbf{A}\mathbf{C}^{-1}\mathbf{r}_n$ transformation.

D. The preconditioned conjugate residual with optimal trial vectors method

In Sec. II C, we have introduced preconditioning to the CR algorithm. In this section, we precondition the CROP algorithm. We use the CROP algorithm in the Eq. (57) as this is most commonly used.

Since the CROP algorithm uses Eq. (57) to determine the expansion coefficients, we have to calculate elements of the $\mathbf{B}$ matrix on the $\mathbf{Y}$ basis. For example, the element for $i,j < n+1$ becomes

$$B_{ij} = (\mathbf{r}_i^{\mathcal{P}})^T\mathbf{r}_j^{\mathcal{P}} = (\mathcal{P}^T\mathbf{r}_i)^T\mathcal{P}^T\mathbf{r}_j = \mathbf{r}_i^T\mathcal{P}\mathcal{P}^T\mathbf{r}_j = \mathbf{r}_i^T\mathbf{C}^{-1}\mathbf{r}_j, \tag{68}$$

where we have used Eq. (63). Equation (58) takes the form

$$B_{ij} = \begin{cases} \mathbf{r}_i^T\mathbf{C}^{-1}\mathbf{r}_j & \text{if } i,j \leqslant n \\ \mathbf{r}_i^T\mathbf{C}^{-1}\tilde{\mathbf{r}}_{n+1} & \text{if } i \leqslant n, j = n+1 \\ \tilde{\mathbf{r}}_{n+1}^T\mathbf{C}^{-1}\mathbf{r}_j & \text{if } i = n+1, j \leqslant n \\ \tilde{\mathbf{r}}_{n+1}^T\mathbf{C}^{-1}\tilde{\mathbf{r}}_{n+1} & \text{if } i = j = n+1. \end{cases} \tag{69}$$

The next trial vector on the $\mathbf{Y}$ basis may, according to Eq. (38), be written as

$$\tilde{\mathbf{Y}}_{n+1} = \mathbf{Y}_n + \mathbf{r}_n. \tag{70}$$

Using $\mathbf{Y} = \mathcal{P}^{-1}\mathbf{x}$ and Eq. (63), we can backtransform Eq. (70) to the following $\mathbf{x}$ basis:

$$\tilde{\mathbf{x}}_{n+1} = \mathbf{x}_n + \mathcal{P}\mathcal{P}^T\mathbf{r}_n = \mathbf{x}_n + \mathbf{C}^{-1}\mathbf{r}_n. \tag{71}$$

In our implementation using the reduced expressions of Eq. (54), we store the vectors $\{\mathbf{x}_{n-1}, \mathbf{x}_n, \tilde{\mathbf{x}}_{n+1}\}$ and the unpreconditioned residuals $\{\mathbf{r}_{n-1}, \mathbf{r}_n, \tilde{\mathbf{r}}_{n+1}\}$. The preconditioner is instead directly used when $\mathbf{B}$ is determined in Eq. (69) and

when a new trial vector is calculated using Eq. (71). Note that it is only when the matrix **B** includes the preconditioner as in Eq. (69) that the results of using the three-dimensional basis $\{\mathbf{x}_{n-1},\mathbf{x}_n,\tilde{\mathbf{x}}_{n+1}\}$ and the full basis are identical.

III. MINIMAL RESIDUAL METHODS FOR NONLINEAR EQUATIONS

We now turn our attention to the solution of sets of nonlinear equations. For a vector-value function $\mathbf{V}=\mathbf{V}(\mathbf{x})$, we are thus interested in finding a vector $\mathbf{x}^\star$ such that

$$\mathbf{V}(\mathbf{x}^\star)=\mathbf{0}. \tag{72}$$

The solution vector $\mathbf{x}^\star$ may be determined using an iterative procedure. At iteration n, we have the approximate solution $\mathbf{x}_n$, and at iteration $n+1$, the approximate solution may be expressed as follows:

$$\mathbf{x}_{n+1}=\mathbf{x}_n+\Delta\mathbf{x}_n. \tag{73}$$

The vector function may be expanded around $\mathbf{x}_n$ as

$$\mathbf{V}(\mathbf{x}_{n+1})=\mathbf{V}(\mathbf{x}_n)+\mathbf{V}^{(1)}(\mathbf{x}_n)\Delta\mathbf{x}_n+\mathcal{O}(\|\Delta\mathbf{x}_n\|^2), \tag{74}$$

where $\mathbf{V}^{(1)}(\mathbf{x}_n)$ is the Jacobian. For a linear vector function, the determination of $\mathbf{x}^\star$ is equivalent to solving linear equations. For a nonlinear expansion, the residual is defined here in terms of the linearized form of $\mathbf{V}(\mathbf{x})$

$$\mathbf{r}(\mathbf{x}_n)=\mathbf{V}(\mathbf{x}_n)+\mathbf{V}^{(1)}(\mathbf{x}_n)\Delta\mathbf{x}_n. \tag{75}$$

$\Delta\mathbf{x}_n$ may be determined by minimizing the residual norm

$$\frac{\partial}{\partial\Delta\mathbf{x}_n}\mathbf{r}^T(\mathbf{x}_{n+1})\mathbf{r}(\mathbf{x}_{n+1})=2\mathbf{V}^{(1)}(\mathbf{x}_n)^T(\mathbf{V}(\mathbf{x}_n)+\mathbf{V}^{(1)}(\mathbf{x}_n)\Delta\mathbf{x}_n)$$
$$=\mathbf{0}, \tag{76}$$

which is satisfied when the Newton equations are solved

$$\mathbf{V}(\mathbf{x}_n)+\mathbf{V}^{(1)}(\mathbf{x}_n)\Delta\mathbf{x}_n=\mathbf{0}. \tag{77}$$

Minimizing the residual norm leads to a quadratically convergent iterative scheme.

Each iteration of the minimum residual scheme requires that a set of linear equations [Eq. (77)] are solved. Although the set of linear equations may be solved using direct methods without explicit construction of $\mathbf{V}^{(1)}$, this is not an efficient approach for most quantum chemical problems. Significantly, more efficient methods may be obtained using quasi-Newton methods where the exact Jacobian is replaced by an approximate Jacobian.

In the next subsection, we show that the DIIS algorithm may be viewed as a quasi-Newton method. Further, if optimal solution vectors are stored in the DIIS algorithm, a generalization of the CROP algorithm is obtained.

A. The DIIS method for nonlinear equations

Let us assume we have carried out an iterative procedure and at iteration n have stored $\tilde{\mathbf{x}}^n:\{\tilde{\mathbf{x}}_0,\tilde{\mathbf{x}}_1,\tilde{\mathbf{x}}_2,\ldots,\tilde{\mathbf{x}}_n\}$ and $\tilde{\mathbf{V}}^n:\{\mathbf{V}(\tilde{\mathbf{x}}_0),\mathbf{V}(\tilde{\mathbf{x}}_1),\mathbf{V}(\tilde{\mathbf{x}}_2),\ldots,\mathbf{V}(\tilde{\mathbf{x}}_n)\}$. The optimal solution in the subspace $\tilde{\mathbf{x}}^n$ may be parametrized as

$$\mathbf{x}_n=\tilde{\mathbf{x}}_n+\sum_{i=0}^{n-1}c_i(\tilde{\mathbf{x}}_i-\tilde{\mathbf{x}}_n), \tag{78}$$

and may be found by minimizing the norm of the residual

$$\mathbf{r}(\mathbf{x}_n)=\mathbf{V}(\mathbf{x}_n)+\mathbf{V}^{(1)}(\mathbf{x}_n)\sum_{i=1}^{n-1}c_i(\mathbf{x}_i-\mathbf{x}_n). \tag{79}$$

Expanding $\mathbf{V}(\mathbf{x})$ as

$$\mathbf{V}(\tilde{\mathbf{x}}_i)=\mathbf{V}(\tilde{\mathbf{x}}_n)+\mathbf{V}^{(1)}(\tilde{\mathbf{x}}_n)(\tilde{\mathbf{x}}_i-\tilde{\mathbf{x}}_n)+\mathcal{O}(\|\tilde{\mathbf{x}}_i-\tilde{\mathbf{x}}_n\|^2) \tag{80}$$

and truncating after linear terms gives the quasi-Newton condition

$$\mathbf{V}(\tilde{\mathbf{x}}_i)-\mathbf{V}(\tilde{\mathbf{x}}_n)=\mathbf{V}^{(1)}(\tilde{\mathbf{x}}_i)(\tilde{\mathbf{x}}_i-\tilde{\mathbf{x}}_n). \tag{81}$$

Using the quasi-Newton condition, the residual in Eq. (79) may be written as

$$\mathbf{r}(\mathbf{x}_n)=\mathbf{V}(\tilde{\mathbf{x}}_n)+\sum_{i=0}^{n-1}c_i(\mathbf{V}(\tilde{\mathbf{x}}_i)-\mathbf{V}(\tilde{\mathbf{x}}_n))$$
$$=\left(1-\sum_{i=0}^{n-1}c_i\right)\mathbf{V}(\tilde{\mathbf{x}}_n)+\sum_{i=1}^{n-1}c_i\mathbf{V}(\tilde{\mathbf{x}}_i)=\sum_{i=0}^{n}c_i\mathbf{V}(\tilde{\mathbf{x}}_i), \tag{82}$$

where we have introduced the parameter c_n and introduced the constraint

$$\sum_{i=0}^{n}c_i=1. \tag{83}$$

Minimizing the residual norm $\mathbf{r}^T(\mathbf{x}_n)\mathbf{r}(\mathbf{x}_n)$ with respect to the parameters $c_i,i=0,1,\ldots,n$ with the constraint Eq. (83) represents a standard DIIS determination of the $\{c_i\}$ coefficients considering $\{\mathbf{V}(\tilde{\mathbf{x}}_0),\mathbf{V}(\tilde{\mathbf{x}}_1),\ldots,\mathbf{V}(\tilde{\mathbf{x}}_n)\}$ as error vectors.[15,16] Using an undetermined Lagrange multiplier to ensure that the constraint Eq. (83) is fulfilled, the expansion coefficients in DIIS are determined from the equations

$$\begin{pmatrix}\mathbf{B}^{\mathrm{DIIS}} & -\mathbf{1}\\ -\mathbf{1} & 0\end{pmatrix}\begin{pmatrix}\mathbf{c}\\ \lambda\end{pmatrix}=\begin{pmatrix}\mathbf{0}\\ -1\end{pmatrix}, \tag{84}$$

where

$$B_{ij}^{\mathrm{DIIS}}=\mathbf{V}^T(\tilde{\mathbf{x}}_i)\mathbf{V}(\tilde{\mathbf{x}}_j),\quad i,j\leqslant n. \tag{85}$$

The preliminary solution at iteration $n+1$ may be expressed as

$$\tilde{\mathbf{x}}_{n+1}=\mathbf{x}_n+\mathbf{r}(\mathbf{x}_n), \tag{86}$$

where $\mathbf{r}(\mathbf{x}_n)$ is obtained from Eq. (82) with the coefficients determined from Eq. (84). The DIIS iterative procedure is established by adding $\tilde{\mathbf{x}}_{n+1}$ and $\mathbf{V}(\tilde{\mathbf{x}}_{n+1})$ to the subspaces $\tilde{\mathbf{x}}^n$ and $\tilde{\mathbf{V}}^n$, respectively. Within the minimal residual framework, the DIIS algorithm may therefore be viewed as a quasi-Newton method where the quasi-Newton condition is applied to the $\tilde{\mathbf{x}}^n$ subspace.

For the case where the expansion of $\mathbf{V}(\mathbf{x})$ contains only linear terms, the DIIS algorithm gives the same result as the CROP parametrization of the CR algorithm. To see this, Eq. (86) may be compared to Eq. (38). In both cases, $\tilde{\mathbf{x}}_{n+1}$ con-

sists of the approximate solution $\mathbf{x}_n$ with the residual $\mathbf{r}(\mathbf{x}_n)$ added. In Eq. (86), $\mathbf{x}_n$ is the optimal solution in the subspace $\{\tilde{\mathbf{x}}_0,\tilde{\mathbf{x}}_1,\ldots,\tilde{\mathbf{x}}_{n-1},\tilde{\mathbf{x}}_n\}$, while $\mathbf{x}_n$ in Eq. (38) is the optimal solution in the subspace $\{\mathbf{x}_0,\mathbf{x}_1,\ldots,\mathbf{x}_{n-1},\tilde{\mathbf{x}}_n\}$ [in Eq. (51) the subspace is given for $\mathbf{x}_{n+1}$]. These two subspaces span the same space, and the same solution $\mathbf{x}_n$ is therefore obtained as long as no truncations are carried out in the subspaces. The DIIS and the CROP algorithms therefore give identical iteration sequences.

When a truncation is carried out in the subspaces, different results are obtained whether it is performed in the $\{\tilde{\mathbf{x}}_0,\tilde{\mathbf{x}}_1,\ldots,\tilde{\mathbf{x}}_{n-1},\tilde{\mathbf{x}}_n\}$ or ln the $\{\mathbf{x}_0,\mathbf{x}_1,\ldots,\mathbf{x}_{n-1},\tilde{\mathbf{x}}_n\}$ subspace. For an expansion $\mathbf{V}(\mathbf{x})$ with only linear terms and a symmetric Jacobian, all except the three last vectors may be discarded in the $\{\mathbf{x}_0,\mathbf{x}_1,\ldots,\mathbf{x}_{n-1},\tilde{\mathbf{x}}_n\}$ subspace, still giving the same $\mathbf{x}_n$ value, as shown in Sec. II B for the CROP method.

For the subspace $\{\mathbf{x}_0,\mathbf{x}_1,\ldots,\mathbf{x}_{n-1},\tilde{\mathbf{x}}_n\}$, we also have that the vector norms in the transformed subspace $\|\mathbf{r}(\mathbf{x}_i)\|$ decrease for increasing i when $\mathbf{V}(\tilde{\mathbf{x}})$ is a linear function. Contrary in the subspace $\{\tilde{\mathbf{x}}_0,\tilde{\mathbf{x}}_1,\ldots,\tilde{\mathbf{x}}_{n-1},\tilde{\mathbf{x}}_n\}$, the vector norms in the transformed subspace $\|\mathbf{V}(\tilde{\mathbf{x}})\|$ may increase for increasing i. Trial vectors $\tilde{\mathbf{x}}_i$ with large values of i are therefore not necessarily better approximations to $\mathbf{x}^\star$ than those with lower i. This has serious consequences for the convergence and no truncations can be carried out in the subspace $\{\tilde{\mathbf{x}}_0,\tilde{\mathbf{x}}_1,\ldots,\tilde{\mathbf{x}}_{n-1},\tilde{\mathbf{x}}_n\}$ without having dramatic consequences for the convergence of the iterative scheme. Summarizing, it may be an advantage to set up an iteration algorithm where the subspace $\{\mathbf{x}_0,\mathbf{x}_1,\ldots,\mathbf{x}_{n-1},\tilde{\mathbf{x}}_n\}$ rather than $\{\tilde{\mathbf{x}}_0,\tilde{\mathbf{x}}_1,\ldots,\tilde{\mathbf{x}}_{n-1},\tilde{\mathbf{x}}_n\}$ is used. We describe in the next subsection, the CROP generalization of the DIIS algorithm where this is performed.

B. The conjugate residual with optimal trial vector algorithm for nonlinear equations

Let us assume we have carried out an iterative procedure where the subspace is built from the optimal vectors $\mathbf{x}_i$. In iteration n, we have thus available the subspaces: $\mathbf{x}^n:\{\mathbf{x}_0,\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n\}$ and the transformed subspace $\mathbf{r}^n:\{\mathbf{r}(\mathbf{x}_0),\mathbf{r}(\mathbf{x}_1),\mathbf{r}(\mathbf{x}_2),\ldots,\mathbf{r}(\mathbf{x}_n)\}$. Our next vector is given as

$$\tilde{\mathbf{x}}_{n+1}=\mathbf{x}_n+\mathbf{r}(\mathbf{x}_n). \tag{87}$$

From $\tilde{\mathbf{x}}_{n+1}$, we may calculate the vector $\mathbf{V}(\tilde{\mathbf{x}}_{n+1})$. We next introduce the intermediate subspace: $\bar{\mathbf{x}}^{n+1}:\{\mathbf{x}_0,\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n,\tilde{\mathbf{x}}_{n+1}\}$ and the corresponding transformed subspace $\bar{\mathbf{r}}^{n+1}:\{\mathbf{r}(\mathbf{x}_0),\mathbf{r}(\mathbf{x}_1),\mathbf{r}(\mathbf{x}_2),\ldots,\mathbf{r}(\mathbf{x}^n),\mathbf{V}(\tilde{\mathbf{x}}_{n+1})\}$. We may proceed finding the optimal solution in the $\bar{\mathbf{x}}^{n+1}$ subspace using an outline similar to the one in the previous section, where the optimal solution vector is parametrized as

$$\mathbf{x}_{n+1}=\sum_{i=0}^{n}c_i\mathbf{x}_i+c_{n+1}\tilde{\mathbf{x}}_{n+1}, \tag{88}$$

with the constraint

$$\sum_{i=0}^{n+1}c_i=1, \tag{89}$$

and where the coefficients are determined by minimizing the norm of the residual

$$\mathbf{r}(\mathbf{x}_{n+1})=\sum_{i=0}^{n}c_i\mathbf{r}(\mathbf{x}_i)+c_{n+1}\mathbf{V}(\tilde{\mathbf{x}}_{n+1}). \tag{90}$$

The expansion coefficients in Eq. (90) may be determined by solving the linear equation in Eq. (57), where $\mathbf{B}^{\mathrm{CR}}$ is replaced by

$$B_{ij}^{\mathrm{CROP}}=\begin{cases}\mathbf{r}^T(\mathbf{x}_i)\mathbf{r}(\mathbf{x}_j) & \text{if } i,j\leqslant n\\ \mathbf{r}^T(\mathbf{x}_i)\mathbf{V}(\tilde{\mathbf{x}}_{n+1}) & \text{if } i\leqslant n,\ j=n+1\\ \mathbf{V}^T(\tilde{\mathbf{x}}_{n+1})\mathbf{r}(\mathbf{x}_j) & \text{if } i=n+1,\ j\leqslant n\\ \mathbf{V}^T(\tilde{\mathbf{x}}_{n+1})\mathbf{V}(\tilde{\mathbf{x}}_{n+1}) & \text{if } i=j=n+1.\end{cases} \tag{91}$$

After having calculated $\mathbf{x}_{n+1}$ and $\mathbf{r}(\mathbf{x}_{n+1})$, these may be added to the $\mathbf{x}^n$ and $\mathbf{r}^n$ subspaces and the iterative procedure is established.

It may appear to be an undesirable feature of the above algorithm that residuals $\mathbf{r}(\mathbf{x}_i)$ and not the exact vector function $\mathbf{V}(\mathbf{x}_i)$ are used in Eq. (90). However, for a nonlinear vector function, terms that are quadratic in $\mathbf{x}_i-\mathbf{x}_n$ are anyhow neglected when the vector function $\mathbf{V}$ is linearized and the quasi-Newton condition is applied. It is also important to note that for the point of expansion $\tilde{\mathbf{x}}_{n+1}$ in the intermediate subspace, the exact vector function is used. At this point, it may therefore be established if the iteration sequence is converged.

The vectors in the transformed subspace are obtained using a norm-minimization algorithm, and the norms of the transformed vectors $\|\mathbf{r}(\mathbf{x}_i)\|$ therefore decrease for increasing i for linear vector functions. Furthermore, as described previously, the algorithm reduces to the CROP parametrization of the CR algorithm for linear vector functions. For vector functions with small nonlinearities and a nearly symmetric Jacobian, only a limited loss of convergence may therefore be expected by reducing the subspace to the three-dimensional subspace containing $\mathbf{x}_{n-1}$, $\mathbf{x}_n$, $\tilde{\mathbf{x}}_{n+1}$. Note, however, that the subspace and transformed subspace vectors do not satisfy conjugacy relations even for a vector function with a small nonlinearity, and that the subspace equations representing the solution of Eq. (88) [or Eq. (88) in a truncated form] therefore must be solved explicitly.

The CROP algorithm has the same form for sets of linear and nonlinear equations. The only difference is in the last element of the intermediate subspaces where the linear or nonlinear transformation is explicitly carried out. For the rest of the elements in the subspace, a linear mixing is carried out of the transformed vectors.

To improve convergence, all calculations should be carried out by applying a preconditioner. In the CROP algorithm, the application of the preconditioner [Eq. (63)] modifies the subspace equations in terms of the elements of the $\mathbf{B}$ matrix, that according to Eq. (69) becomes

$$B_{ij} = \mathbf{r}_i^T \mathbf{C}^{-1} \mathbf{r}_j. \tag{92}$$

New trial vectors are obtained, according to Eq. (71), as

$$\tilde{\mathbf{x}}_{n+1} = \mathbf{x}_n + \mathbf{C}^{-1}\mathbf{r}(\mathbf{x}_n). \tag{93}$$

IV. CONVERGENCE RATE OF THE CONJUGATE RESIDUAL ALGORITHM

We now examine the convergence rate of the CR method and compare it to the one of the CG method. The CR and CROP methods are mathematically identical for linear equations, and the CROP method is therefore not considered in this subsection. Although the CR algorithm converges for general symmetric matrices, in the following, it will be assumed that **A** is symmetric and positive definite.

A. The CG and CR methods as error-minimization methods

A measure of the error of the approximate solution $\mathbf{x}_n$ to Eq. (1) is the error vector

$$\mathbf{e}_n = \mathbf{x}_n - \mathbf{x}^\star, \tag{94}$$

where $\mathbf{x}^\star$ is the exact solution. In the nth iteration of the CG method, the function $\frac{1}{2}\mathbf{x}_n^T\mathbf{A}\mathbf{x}_n - \mathbf{x}_n^T\mathbf{b}$ is minimized.[4,5] This minimization may be expressed in terms of the error vector as

$$\tfrac{1}{2}\mathbf{x}_n^T\mathbf{A}\mathbf{x}_n - \mathbf{x}_n^T\mathbf{b} = \tfrac{1}{2}\mathbf{e}_n^T\mathbf{A}\mathbf{e}_n - \tfrac{1}{2}\mathbf{x}^{\star T}\mathbf{b}. \tag{95}$$

Introducing the Frobenius norm for a vector $\mathbf{x}$ and a symmetric and positive definite matrix $\mathbf{M}$,

$$\|\mathbf{x}\|_M = \sqrt{\mathbf{x}^T\mathbf{M}\mathbf{x}}, \tag{96}$$

it is seen that the CG method in iteration n minimizes the norm $\|\mathbf{e}_n\|_\mathbf{A}$.

The residual may be expressed in terms of the error vector as

$$-\mathbf{r}_n = \mathbf{A}\mathbf{e}_n. \tag{97}$$

The CR method minimizes in iteration n the gradient norm $\|\mathbf{r}_n\|$, which using Eqs. (96) and (97) may be rewritten as

$$\|\mathbf{r}_n\| = \sqrt{(\mathbf{A}\mathbf{e}_n)^T(\mathbf{A}\mathbf{e}_n)} = \|\mathbf{e}_n\|_{\mathbf{A}^2}, \tag{98}$$

showing that the CR method in iteration n minimizes the norm $\|\mathbf{e}_n\|_{\mathbf{A}^2}$. The CG and the CR methods thus minimize various vector space norms

$$\|\mathbf{e}_n\|_{\mathbf{A}^q} \begin{cases} q=1 & \text{for CG} \\ q=2 & \text{for CR}. \end{cases} \tag{99}$$

In both the CG and CR methods, the approximate solution $\mathbf{x}_n$ may be written as

$$\mathbf{x}_n = \mathbf{x}_0 + \sum_{i=0}^{n-1} \gamma_i \mathbf{A}^i \mathbf{r}_0, \tag{100}$$

where the expansion coefficients are determined by minimizing the appropriate vector space norm. The vector $\mathbf{x}_n - \mathbf{x}_0$ is a vector in the Krylov space $\mathcal{K}_n(\mathbf{A},\mathbf{r}_0) = \text{span}\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \mathbf{A}^2\mathbf{r}_0, \ldots, \mathbf{A}^{n-1}\mathbf{r}_0\}$. Using Eq.(100), the error vector in iteration n may be written as a polynomial $\mathbf{P}_n$ of the order n in the matrix $\mathbf{A}$ times the error vector of the initial iteration

$$\mathbf{e}_n = \mathbf{P}_n(\mathbf{A})\mathbf{e}_0 = \left(1 - \sum_{i=1}^{n} \gamma_i \mathbf{A}^i\right)\mathbf{e}_0. \tag{101}$$

From Eq. (101), it is seen that

$$\mathbf{P}_n(\mathbf{0}) = \mathbf{1}. \tag{102}$$

The polynomial $\mathbf{P}_n$ is determined in the various iterations of the CG or CR algorithm. In iteration n, the norm $\|\mathbf{e}_n\|_{\mathbf{A}^q}$ is minimized, and we may therefore consider the polynomial to be the solution of the variational problem

$$\|\mathbf{e}_n\|_{\mathbf{A}^q} = \min_{\mathbf{P}_n} \|\mathbf{P}_n(\mathbf{A})\mathbf{e}_0\|_{\mathbf{A}^q}, \tag{103}$$

with the constraint Eq. (102).

To proceed, we introduce the spectral resolution of the matrix $\mathbf{A}$ in terms of eigenvalues α_k and orthonormal eigenvectors $\mathbf{v}_k$

$$\mathbf{A} = \sum_{k=1}^{d} \alpha_k \mathbf{v}_k \mathbf{v}_k^T, \tag{104}$$

$$\mathbf{v}_k^T \mathbf{v}_l = \delta_{kl}. \tag{105}$$

Writing the initial error vector on the eigenvector basis

$$\mathbf{e}_0 = \sum_{k=1}^{d} \xi_k \mathbf{v}_k, \tag{106}$$

the vector $\mathbf{A}^i\mathbf{e}_0$ may be expressed as

$$\mathbf{A}^i\mathbf{e}_0 = \sum_{k=1}^{d} \alpha_k^i \xi_k \mathbf{v}_k, \tag{107}$$

where we have used Eqs. (104) and (105). Using Eq. (101), the error vector of iteration n may, in the spectral representation, be expressed as

$$\mathbf{e}_n = \sum_{k=1}^{d} P_n(\alpha_k)\xi_k \mathbf{v}_k, \tag{108}$$

where $P_n(\alpha_k)$ is the scalar polynomial corresponding to the matrix polynomial $\mathbf{P}_n(\mathbf{A})$ and therefore satisfies

$$P(0) = 1. \tag{109}$$

From Eq. (103), it is seen that P_n may be defined as the polynomial that is the solution to the minimization problem

$$\|\mathbf{e}_n\|_{\mathbf{A}^q}^2 = \min_{P_n} \left(\sum_{k=1}^{d} (P_n(\alpha_k))^2 \alpha_k^q \xi_k^2 \right) \tag{110}$$

with the constraint Eq. (109).

B. Convergence of the CR and CG methods

From Eq. (110), it is seen that the convergence of an iteration sequence is very dependent on the number of different eigenvalues of **A**. As α_k and ξ_k^2 are positive quantities, convergence is obtained when the polynomial $P_n(\alpha_k)$ has all

eigenvalues as roots. As a polynomial of the order m has at most m roots, convergence is obtained when the order of the polynomial P_n, (the number of iterations) is equal to the number of different eigenvalues of $\mathbf{A}$. The CG and the CR methods therefore converge fast when there is only a few distinct eigenvalues.

The number of eigenvalues of $\mathbf{A}$ that are different is usually much larger than the number of iterations and it, therefore, becomes important to estimate the rate of convergence of the iteration sequence. To this end, we first note that if $\aleph$ is the set of all eigenvalues of $\mathbf{A}$ then for arbitrary polynomials P_n we have

$$\sum_{k=1}^{d}(P_n(\alpha_k))^2\alpha_k^q\xi_k^2 \leqslant \max_{\alpha\in\aleph}(P_n(\alpha))^2\sum_{k=1}^{d}\alpha_k^q\xi_k^2 = \max_{\alpha\in\aleph}(P_n(\alpha))^2\|\mathbf{e}_o\|_{\mathbf{A}^q}, \tag{111}$$

where α is the eigenvalue for which the polynomial P_n has its largest eigenvalue and Eqs. (104) and (106) are used to identify $\|\mathbf{e}_0\|_{\mathbf{A}^q}^2$. Inserting Eq. (111) in Eq. (110) gives an upper bound for the norm of the error vector

$$\|\mathbf{e}_n\|_{\mathbf{A}^q}^2 \leqslant \min_{P_n}\max_{\alpha\in\aleph}(P_n(\alpha))^2\|\mathbf{e}_0\|_{\mathbf{A}^q}^2, \quad \text{where } P_n(0)=1. \tag{112}$$

The polynomial that minimizes the error-norm of Eq. (112) is not known. To obtain an estimate of it, we replace the above minimax problem for a discrete set of parameters $\aleph$ with a minimax problem for a variable α defined on the interval between the smallest and largest eigenvalue of $\mathbf{A}$, $\alpha_{\min}$, $\alpha_{\max}$,

$$\frac{\|\mathbf{e}_n\|_{\mathbf{A}^q}}{\|\mathbf{e}_0\|_{\mathbf{A}^q}} \leqslant \min_{P_n}\max_{\alpha\in[\alpha_{\min},\alpha_{\max}]}|P_n(\alpha)|. \tag{113}$$

This replacement is feasible if the dimension of the matrix is large and the eigenvalues are evenly distributed in the interval between $\alpha_{\min}$ and $\alpha_{\max}$. In the following, we will assume that these conditions are fulfilled.

The polynomials that have the smallest maximum numerical value in a given interval are the so-called Chebyshev polynomials of the first degree or just the Chebyshev polynomials and they are well studied in the mathematical literature.[24] We give a brief introduction to these. The Chebyshev polynomials $T_n(x)$ are defined in the interval $[-1,1]$ in terms of trigonometric functions

$$T_n(x)=\cos(n\arccos(x)). \tag{114}$$

In the interval $[0,\pi]$, where $\arccos x$ is defined, it may easily be shown

$$\cos(n\theta)=\tfrac{1}{2}[(\cos\theta+\sqrt{\cos^2\theta-1})^n + (\cos\theta+\sqrt{\cos^2\theta-1})^{-n}], \tag{115}$$

using $\exp(i\theta)=\cos\theta+\sqrt{\cos^2\theta-1}$ and $\cos n\theta=\frac{1}{2}[\exp(in\theta)+\exp(-in\theta)]$. Inserting Eq. (115) in Eq. (114) and setting $\theta=\arccos x$ gives the following closed form for the Chebyshev polynomials:

$$T_n(x)=\tfrac{1}{2}[(x+\sqrt{x^2-1})^n+(x+\sqrt{x^2-1})^{-n}]. \tag{116}$$

The most important feature of the Chebyshev polynomials in the present context is their minimax property which may be stated as follows.[6] Consider polynomials $P_n(x)$ of order n defined in an interval of $-1\leqslant x\leqslant 1$ having a fixed predefined value C at a given point x_C, i.e., $P_n(x_C)=C$. Among these polynomials, the polynomial which has the smallest largest numerical value is the scaled Chebyshev polynomial, cT_n, where c is a constant that is determined by the predefined value of the polynomial.

To apply the minimax property of the Chebyshev polynomials on the minimax problem in Eq. (113), we first have to carry out a variable substitution where the variable α defined in the interval $[\alpha_{\min},\alpha_{\max}]$ is substituted with another variable defined in the interval $[-1,1]$. This variable substitution may be carried out introducing

$$\omega=\frac{\alpha_{\max}+\alpha_{\min}-2\alpha}{\alpha_{\max}-\alpha_{\min}}, \tag{117}$$

where ω goes from 1 to -1 when α goes from $\alpha_{\min}$ to $\alpha_{\max}$. The polynomial $P_n(\alpha)$ may therefore be expressed as

$$P_n(\alpha)=cT_n\left(\frac{\alpha_{\max}+\alpha_{\min}-2\alpha}{\alpha_{\max}-\alpha_{\min}}\right), \tag{118}$$

where the coefficient c is determined from the requirement that $P_n(0)=1$. The c value therefore becomes

$$c=\frac{1}{T_n\left(\frac{\alpha_{\max}+\alpha_{\min}}{\alpha_{\max}-\alpha_{\min}}\right)}, \tag{119}$$

and the polynomial in Eq. (118) becomes

$$P_n(\alpha)=\frac{T_n\left(\frac{\alpha_{\max}+\alpha_{\min}-2\alpha}{\alpha_{\max}-\alpha_{\min}}\right)}{T_n\left(\frac{\alpha_{\max}+\alpha_{\min}}{\alpha_{\max}-\alpha_{\min}}\right)}. \tag{120}$$

The polynomial in Eq. (120) is thus the polynomial of the order n which has the smallest maximum value in the interval $\alpha_{\min}\leqslant\alpha\leqslant\alpha_{\max}$. Inserting this polynomial in Eq. (113) gives

$$\frac{\|\mathbf{e}_n\|_{\mathbf{A}^q}}{\|\mathbf{e}_0\|_{\mathbf{A}^q}} \leqslant \max_{\alpha\in[\alpha_{\min},\alpha_{\max}]}\left|\frac{T_n\left(\frac{\alpha_{\max}+\alpha_{\min}-2\alpha}{\alpha_{\max}-\alpha_{\min}}\right)}{T_n\left(\frac{\alpha_{\max}+\alpha_{\min}}{\alpha_{\max}-\alpha_{\min}}\right)}\right|. \tag{121}$$

From Eq. (114), it is seen that the numerical values of the Chebyshev polynomial are less than or equal to 1. $T_n[(\alpha_{\min}+\alpha_{\min}-2\alpha)/(\alpha_{\max}-\alpha_{\min})]$ is therefore less than or equal to 1 for $\alpha\in[\alpha_{\min},\alpha_{\max}]$ and Eq. (121) may therefore be simplified as

$$\frac{\|\mathbf{e}_n\|_{\mathbf{A}^q}}{\|\mathbf{e}_0\|_{\mathbf{A}^q}} \leqslant \left(\left|T_n\left(\frac{\alpha_{\max}+\alpha_{\min}}{\alpha_{\max}-\alpha_{\min}}\right)\right|\right)^{-1}. \tag{122}$$

Introducing the condition number of the matrix

$$\kappa = \frac{\alpha_{\max}}{\alpha_{\min}}, \tag{123}$$

Eq. (122) becomes

$$\frac{\|\mathbf{e}_n\|_{\mathbf{A}^q}}{\|\mathbf{e}_0\|_{\mathbf{A}^q}} \leqslant \left(\left| T_n\left(\frac{\kappa+1}{\kappa-1} \right) \right| \right)^{-1}. \tag{124}$$

Using Eq. (116) for the Chebyshev polynomial gives after a bit of algebra

$$\frac{\|\mathbf{e}_n\|_{\mathbf{A}^q}}{\|\mathbf{e}_0\|_{\mathbf{A}^q}} \leqslant \frac{2}{\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1} \right)^n + \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} \right)^n}. \tag{125}$$

As the term $(\sqrt{\kappa}-1)/(\sqrt{\kappa}+1)$ goes to zero for large values of n, it is standard to neglect this term from the convergence rate, giving the standard expression for the convergence of the CG and CR method

$$\frac{\|\mathbf{e}_n\|_{\mathbf{A}^q}}{\|\mathbf{e}_0\|_{\mathbf{A}^q}} \leqslant 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} \right)^n. \tag{126}$$

We thus conclude that the CG and CR methods have the same rate of convergence for a matrix of large dimensions and with evenly distributed eigenvalues.

In the above, we have analyzed errors in terms of the error vectors $\mathbf{e}_i$, which may be computed only after convergence. The errors may alternatively be expressed in terms of the residuals $\mathbf{r}_i$ [see Eq. (97)] which are computed in each iteration. In the CR algorithm, the norms $\|\mathbf{r}_i\|$ are by definition minimized in each iteration. Using the CG algorithm that minimizes the norm $\|\mathbf{e}_i\|_{\mathbf{A}}$, it is seen that each iteration in this method minimizes the norm $\|\mathbf{r}_i\|_{\mathbf{A}^{-1}}$, which is not trivial to compute.

In actual calculations, significantly faster convergence is often observed than predicted by Eq. (126). In fact, the CG method, in general, converges superlinearly.[25] This arises because the CG algorithm during the iterative process accurately solves the linear equations for the extreme eigenvalues and the effective condition number of the matrix therefore becomes significantly reduced during the iterative procedure.

V. TEST CALCULATIONS

In this section, we present numerical studies of the convergence of the presented algorithms. We consider the second order Møller–Plesset (MP2) amplitude equations in the projected Löwdin orthogonalized atomic orbital (PLAO)[21] basis as an example of linear equations and the coupled-cluster singles doubles (CCSD) amplitude equations also in the PLAO basis as an example of nonlinear equations.[22] For both the MP2 and CCSD amplitude equations, we used the diagonal preconditioner described by Weijo *et al.*[21] The MP2 amplitude equations were solved for C_6H_2 using the cc-pVTZ basis,[26] at the equilibrium geometry obtained at the MP2/cc-pVTZ level. The CCSD amplitude equations were solved for H_2O using the cc-pVDZ basis,[25] at the equilibrium geometry obtained at the MP2/cc-pVTZ level. At the equilibrium geometry, the CCSD amplitude equations have a weak nonlinearity. To investigate the performance for increasing nonlinearity, we consider also calculations for H_2O where for a fixed H-O-H bond angle, the O-H bond distance is stretched to 1.5, 2.0, and 2.5 times its equilibrium value R_e maintaining the C_{2v} geometry.

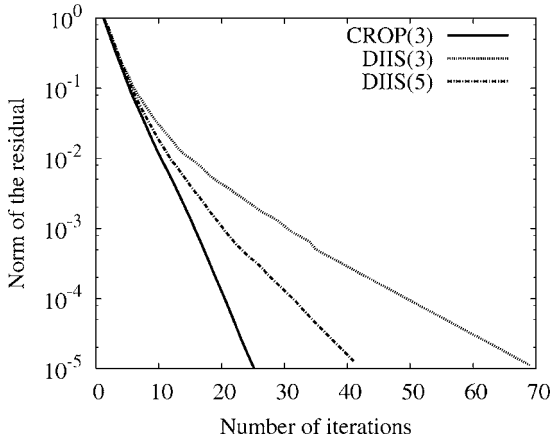

FIG. 1. Convergence of the MP2 equations in the PLAO basis using the CROP algorithm with a three vector history and DIIS with a three and five vector history.

The convergence of the MP2 amplitude equations for C_6H_2 is displayed in Fig. 1 using the CROP algorithm with a three vector subspace, CROP(3). For comparison, the DIIS results are also reported using a subspace of three vectors, DIIS(3), and using five vectors, DIIS(5). The iteration sequence for the CROP(3) calculation is identical to the one obtained without truncation in the subspace. Without truncation in the subspace, DIIS also reproduces the CROP(3) results. However, contrary to the CROP algorithm, the DIIS method gives a significant degradation in performance for a truncated subspace, as seen in Fig. 1. With a three vector subspace, DIIS(3), convergence to a residual norm 10^{-5} thus takes 70 iterations while the CROP(3) calculation uses 24 iterations to obtain the same threshold. A 5 vector subspace, DIIS(5), uses 42 iterations. For sets of linear equations, the CROP algorithm thus preserves the history of all previous iterations with a three vector subspace, whereas a significant degradation is obtained in DIIS when truncation is performed in the subspace.

The convergence of the CCSD amplitude equations at the equilibrium geometry and with the O–H bond lengths stretched to $1.5R_e$, $2.0R_e$, and $2.5R_e$ is given in Fig. 2. The nonlinearity of the amplitude equations increases when the bond length is stretched as reflected by the maximum value of the two electron excitation amplitudes of 1.8×10^{-2}, 2.9×10^{-2}, 5.4×10^{-1}, and 2.2×10^{-1} for the $1.0R_e$, $1.5R_e$, $2.0R_e$, and $2.5R_e$ calculations, respectively. In Fig. 2, we give results for the CROP and DIIS algorithm with no truncation of the subspace and for truncated subspaces. For the bond distances $1.0R_e$, $1.5R_e$, and $2.0R_e$, a three vector subspace, CROP(3), leads to a minor degradation in performance. For the DIIS method, the degradation is large with a three vector subspace, DIIS(3) and still significant with a five vector subspace, DIIS(5). For the O–H bond distance $2.5R_e$, the nonlinearity is large. Keeping all vectors in the CROP algorithm leads to a monotonic but slow convergence, whereas for a 20 vector subspace, CROP(20), convergence could not be ob-

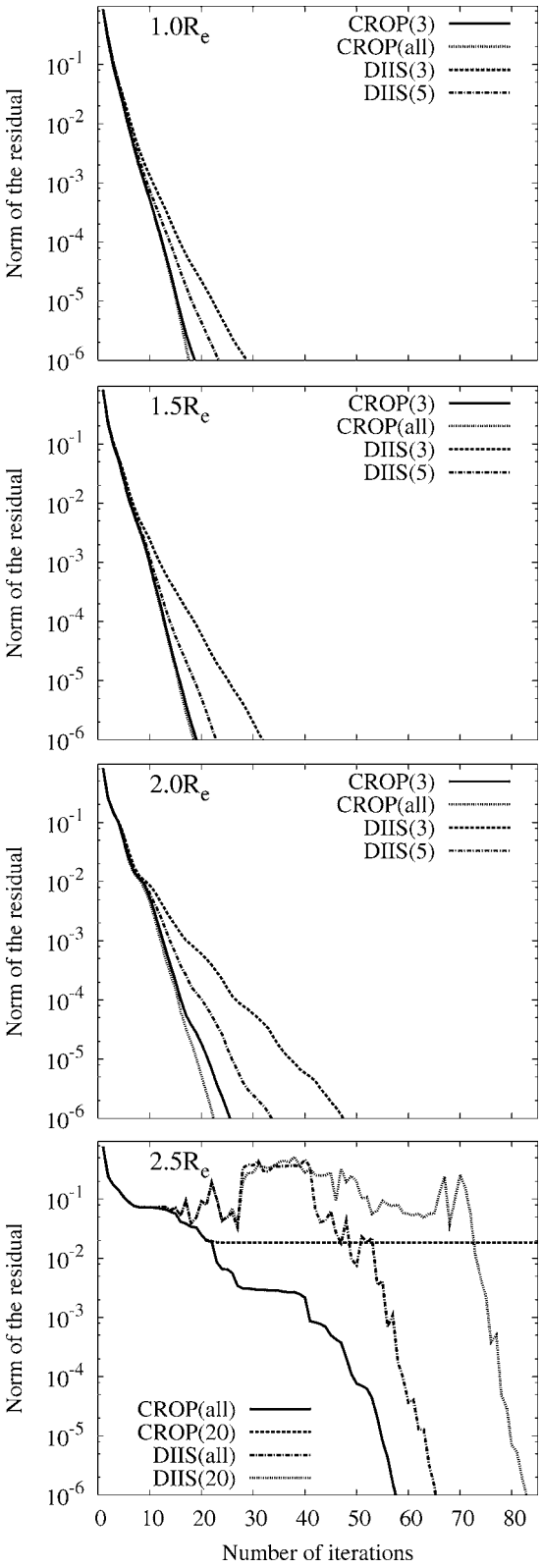

FIG. 2. Convergence of the CCSD equations in the PLAO basis for the H_2O molecule using the CROP and DIIS algorithms with various sizes of the subspace. Four geometries with increasing nonlinearity of the CCSD equations are considered. These geometries correspond to the bond distances of $1.0R_e$, $1.5R_e$, $2.0R_e$, and $2.5R_e$, where R_e is the equilibrium distance at the MP2/cc-pVTZ level of theory.

tained. For the DIIS approach with a full subspace, a rather erratic behavior is observed in the initial 50 iterations after which the DIIS algorithm converges fast. With a 20 vector subspace, DIIS(20), the erratic behavior is extended up to about iteration 70.

To summarize, it appears that for a weak to moderate nonlinearity, it is sufficient to keep a three vector subspace in the CROP algorithm. For a stronger nonlinearity a larger subspace is required. In all cases the CROP algorithm outperformed the DIIS approach.

VI. CONCLUSIONS

Minimal residual algorithms for the solution of linear and nonlinear equations have been examined. In particular, it has been shown that the CR has the same attractive feature as the CG algorithm, namely, that for linear equations only information from the last iteration is required to ensure that the solution is conjugate to the directions of all previous iterations. This feature is used to set up the conjugate residual with optimal trial vectors (CROP) algorithm where optimal trial vectors of each iteration rather than the directions are stored and where information from the last iteration is sufficient to reproduce the CR result. The CR and CG algorithms are shown to have the same convergence characteristics.

The CROP algorithm for linear equations gives the same result as the DIIS algorithm where no truncation is made in the subspace. However, for a truncated subspace, the DIIS algorithm shows a significant degradation of performance as demonstrated by the test calculations for the MP2 amplitude equations. The CROP algorithm thus constitutes an optimal way of preserving the information from the previous iterations.

The use of a subspace containing trial vectors, allows a straightforward extension of the CROP algorithm to nonlinear equations. Test calculations for nonlinear CCSD amplitude equations show that for a weak to moderate nonlinearity, only a minor degradation in performance is obtained with a three vector subspace. For a strong nonlinearity, a larger subspace may be required.

Summarizing, the CROP algorithm constitutes an efficient algorithm for solving nonlinear equations outperforming the DIIS algorithm through a more efficient way of preserving the information from the previous iterations.

ACKNOWLEDGMENTS

This work has been supported by the Lundbeck Foundation and The Danish Natural Science Research Council (Grant No. 21-04-0268).

[1] P. Sałek, S. Høst, L. Thøgersen, P. Jørgensen, P. Manninen, J. Olsen, B. Jansík, S. Reine, F. Pawłowski, E. Tellgren, T. Helgaker, and S. Coriani, J. Chem. Phys. **126**, 114110 (2007).
[2] A. Daniels and G. Scuseria, Phys. Chem. Chem. Phys. **2**, 2173 (2000).
[3] T. Helgaker, P. Jørgensen, and J. Olsen, *Molecular Electronic-Structure Theory* (Willey, Chichester, 2002).
[4] M. R. Hestenes and E. Stiefel, J. Res. Natl. Bur. Stand. **49**, 409 (1952).
[5] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C++: The Art of Scientific Computing*, 2nd ed. (Cambridge University Press, Cambridge, 2002).
[6] J. R. Shewchuk, *An Introduction to the Conjugate Gradient Method With-*

out the Agonizing Pain (Carnegie Mellon University, Pittsburgh, PA, 1994).
[7] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed. (SIAM, Philadelphia, 2003).
[8] J. Pople, R. Krishnan, H. Schlegel, and J. S. Binkley, Int. J. Quantum Chem., Quantum Chem. Symp. **13**, 225 (1979).
[9] P. E. S. Wormer, F. Visser, and J. Paldus, J. Comput. Phys. **48**, 23 (1982).
[10] C. C. Paige and M. A. Saunders, SIAM (Soc. Ind. Appl. Math.) J. Numer. Anal. **12**, 617 (1975).
[11] Y. Saad and M. Schultz, SIAM (Soc. Ind. Appl. Math.) J. Sci. Stat. Comput. **7**, 856 (1986).
[12] E. Stiefel, Comment. Math. Helv. **29**, 157 (1955).
[13] P. Vinsome, Proc. Fourth Symposium on Reservoir Simulation, Society of Petroleum Enginers of AIME 1976 (unpublished), pp. 149–159.
[14] S. C. Eisenstat, H. C. Elman, and M. H. Schultz, SIAM (Soc. Ind. Appl. Math.) J. Numer. Anal. **20**, 345 (1983).
[15] P. Pulay, Chem. Phys. Lett. **73**, 393 (1980).
[16] P. Pulay, J. Comput. Chem. **3**, 556 (1982).
[17] P. Csaszar and P. Pulay, J. Mol. Struct. **114**, 31 (1984).
[18] O. Farkas and H. B. Schlegel, Phys. Chem. Chem. Phys. **11**, 11 (2002).
[19] G. E. Scuseria, T. J. Lee, and H. F. Schaefer, Chem. Phys. Lett. **130**, 236 (1986).
[20] J. Olsen, in *Molecular Quantum Mechanics: Analytic Gradients and Beyond*, edited by A. G. Csaszar, G. Fogarasi, H. F. Schaefer, and P. G. Szalay (ELTE Institute of Chemistry, Budapest, 2007), pp. 85–89.
[21] V. Weijo, P. Manninen, P. Jørgensen, O. Christiansen, and J. Olsen, J. Chem. Phys. **127**, 074106 (2007).
[22] O. Christiansen, P. Manninen, P. Jørgensen, and J. Olsen, J. Chem. Phys. **124**, 084103 (2006).
[23] See EPAPS Document No. E-JCPSA6-128-054821 for induction proofs of the conjugate residual algorithm. For more information on EPAPS, see http://www.aip/pubservs/epaps.html.
[24] G. Arfken, *Mathematical Methods for Physicists*, 3rd ed. (Academic, Orlando, Florida, 1985), pp. 731–748.
[25] A. van der Sluis and H. A. van der Vorst, Numer. Math. **48**, 543 (1986).
[26] J. Thom, H. Dunning, J. Chem. Phys. **90**, 1007 (1989).